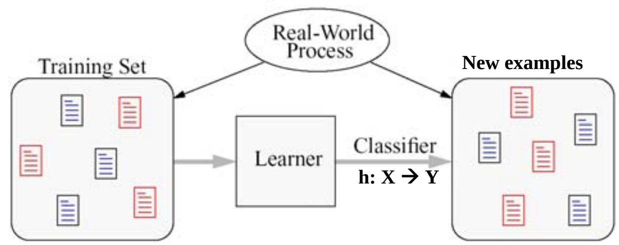
Machine Learning Experiments and Overfitting

CS472/CS473 – Fall 2005

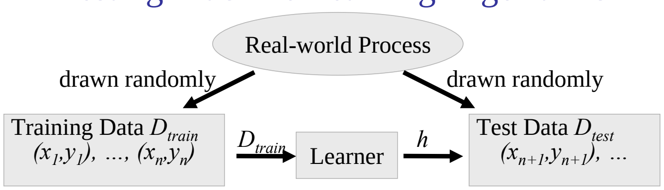
Inductive Learning Setting



Learning as Prediction:

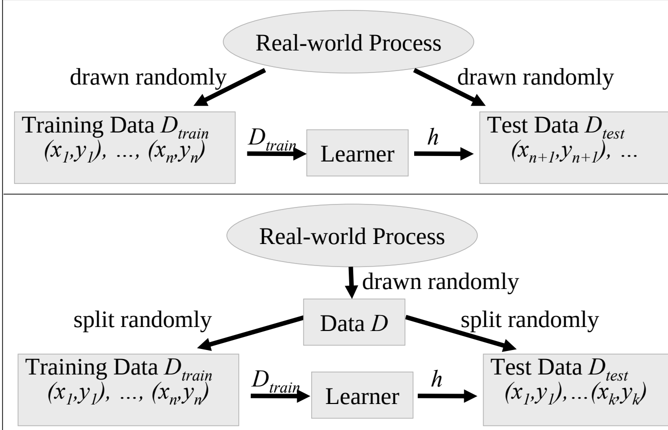
- Learner induces a general rule h from a set of observed examples that classifies new examples accurately.

Testing Machine Learning Algorithms



- **Machine Learning Experiment:**
  - Gather training examples D_{train}
  - Run learning algorithm on D_{train} to produce h
  - Gather Test Examples D_{test}
  - Apply h to D_{test} and measure how many test examples are predicted correctly by h

Test/Training Split



Measuring Prediction Performance

Definition: *The* **training error** $Err_{D_{train}}(h)$ *on training data* $D_{train} = ((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n))$ *of a hypothesis* h *is* $Err_{D_{train}}(h) = \frac{1}{n}\sum_{i=1}^{n} \Delta(h(\vec{x}_i), y_i)$.

Definition: *The* **test error** $Err_{D_{test}}(h)$ *on test data* $D_{test} = ((\vec{x}_1, y_1), \ldots, (\vec{x}_k, y_k))$ *of a hypothesis* h *is* $Err_{D_{test}}(h) = \frac{1}{n}\sum_{i=1}^{k} \Delta(h(\vec{x}_i), y_i)$.

Definition: *The* **prediction/generalization/true error** $Err_P(h)$ *of a hypothesis* h *for a learning task* $P(X,Y)$ *is*

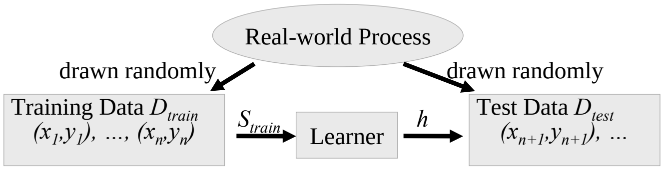
$$Err_P(h) = \sum_{\vec{x} \in X, y \in Y} \Delta(h(\vec{x}), y) P(X = \vec{x}, Y = y).$$

Definition: *The zero/one-loss function* $\Delta(a,b)$ *returns* 1 *if* $a \neq b$ *and* 0 *otherwise.*

Performance Measures

- **Error Rate**
  - Fraction (or percentage) of false predictions
- **Accuracy**
  - Fraction (or percentage) of correct predictions
- **Precision/Recall**
  - Applies only to binary classification problems (classes pos/neg)
  - Precision: Fraction (or percentage) of correct predictions among all examples predicted to be positive
  - Recall: Fraction (or percentage) of correct predictions among all positive examples

Learning as Prediction Task

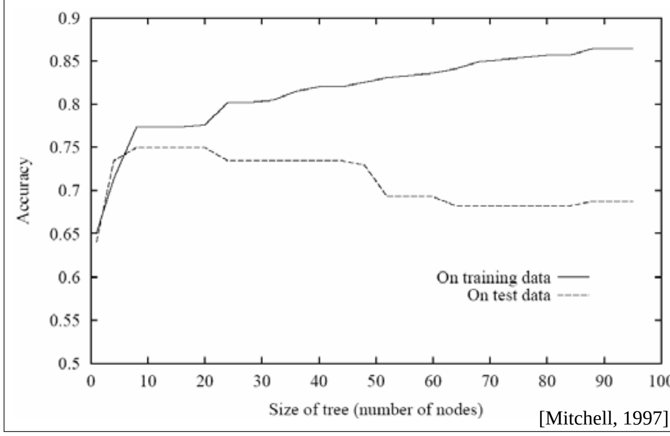


- **Goal:** Find h with small prediction error $Err_P(h)$.
- **Strategy:** Find (any?) h with small error $Err_{D_{train}}(h)$ on D_{train}.

Inductive Learning Hypothesis: Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over any other unobserved examples.

→ **Is Inductive Learning Hypothesis really true?**

Overfitting vs. Underfitting



[Mitchell, 1997]

Example: Text Classification

- **Task: Learn rule that classifies Reuters Business News**
  - Class +: "Corporate Acquisitions"
  - Class -: Other articles
  - 2000 training instances
- **Representation:**
  - Boolean attributes, indicating presence of a keyword in article
  - 9947 such keywords (more accurately, word "stems")

LAROCHE STARTS BID FOR NECO SHARES **+**

Investor David F. La Roche of North Kingstown, R.I., said he is offering to purchase 170,000 common shares of NECO Enterprises Inc at 26 dlrs each. He said the successful completion of the offer, plus shares he already owns, would give him 50.5 pct of NECO's 962,016 common shares. La Roche said he may buy more, and possible all NECO shares. He said the offer and withdrawal rights will expire at 1630 EST/2130 gmt, March 30, 1987.

SALANT CORP 1ST QTR FEB 28 NET **-**

Oper shr profit seven cts vs loss 12 cts. Oper net profit 216,000 vs loss 401,000. Sales 21.4 mln vs 24.9 mln. NOTE: Current year net excludes 142,000 dlr tax credit. Company operating in Chapter 11 bankruptcy.

Text Classification Example: Results

- **Data**
  - Training Sample: 2000 examples
  - Test Sample: 600 examples
- **Full Decision Tree:**
  - Size: 437 nodes Training Error: 0.0% Test Error: 11.0%
- **Early Stopping Tree:**
  - Size: 299 nodes Training Error: 2.6% Test Error: 9.8%

Example: Smart Investing

Task: Pick stock analyst based on past performance.

Experiment:
- Have analyst predict "next day up/down" for 10 days.
- Pick analyst that makes the fewest errors.

Situation 1:
- 1 stock analyst {A1}, A1 makes 5 errors

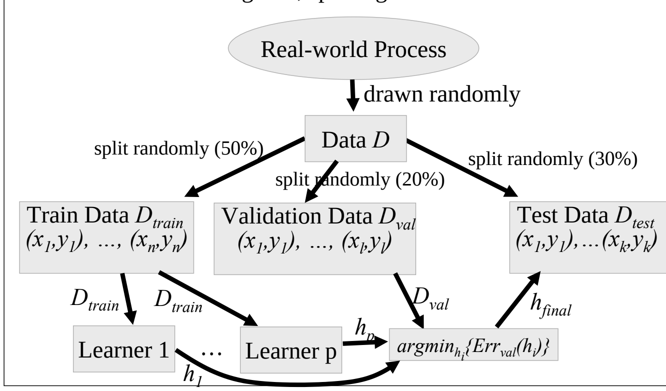
Situation 2:
- 3 stock analysts {A1,B1,B2}, B2 best with 1 error

Situation 3:
- 1003 stock analysts {A1,B1,B2,C1,…,C1000}, C543 best with 0 errors

Which analysts are you most confident in, A1, B2, or C543?

Selecting Algorithm Parameters

Optimal choice of algorithm parameters depends on learning task:
- k in k-nearest neighbor, splitting criterion in decision trees



K-fold Cross Validation

- **Given**
  - Sample of labeled instances D (after putting aside D_{test})
  - Learning Algorithms $A_1 \ldots A_p$ (e.g. k-NN with different k)
- **Compute**
  - Randomly partition D into k equally sized subsets $D_1 \ldots D_k$
  - For i from 1 to k
    - Train $A_1 \ldots A_p$ on $\{D_1 \ldots D_{i-1} D_{i+1} \ldots D_k\}$ and get $h_1 \ldots h_p$.
    - Apply $h_1 \ldots h_p$ to D_i and compute $Err_{D_i}(h_1) \ldots Err_{D_i}(h_p)$.
- **Estimate**
  - $Err_{CV}(A_i) \leftarrow 1/k \sum_{i \in \{1..k\}} Err_{D_i}(h_1)$ is estimate of the prediction error of A_i
  - Pick algorithm A_{best} with lowest $Err_{CV}(A_i)$
  - Train A_{best} on D and output resulting h